In the Claims:

1. (Canceled)

2. (New) A system comprising:

a multithreaded processor comprising a plurality of microengines, a memory controller, a first bus interface and a second bus interface, the second bus interface comprising a first-in-first-out memory with a plurality of elements to store packet data and packet status;

a system bus coupled to the first bus interface;

a network bus coupled to the second bus interface;

a media access control device coupled to the network bus; and

a memory system coupled to the memory controller.

3. (New) The system of Claim 1, wherein the second bus interface comprises a controller and a first-in-first-out memory operable to store data from the memory system to be sent to the second bus.

4. (New) The system of Claim 1, wherein each microengine has a plurality of hardware-controlled threads operable to be active simultaneously.

5. (New) The system of Claim 1, wherein the second bus interface is operable to issue N requests for packet data and packet status to the media access control device coupled to the network bus, each request for M bytes; receive packet data and packet status and store received packet data and packet status in an element of the first-in-first-out memory; detect an end-

of-packet indicator; determine if the last M received bytes contain data or packet status; and if the last M received bytes contain packet data, issue another request of M bytes to retrieve packet status.

6. (New) A processor comprising:

a bus interface coupled to a bus, the bus interface comprising first-in-first-out memories;

a plurality of microengines coupled to the bus interface, each microengine having a plurality of hardware-controlled threads operable to be active simultaneously, the microengines being operable to transfer data to the first-in-first-out memories;

a first memory controller; and

a second memory controller.

7. (New) The processor of Claim 6, wherein bus interface is coupled to a bus, the bus being coupled to a media access control device.

8. (New) The processor of Claim 6, wherein each microengine maintains a plurality of program counters and states associated with the program counters.

9. (New) The processor of Claim 6, wherein a first thread of a microengine is operable to request access to a memory coupled to the first memory controller, and a second thread of the microengine is operable while the first thread waits for data from the memory.

10. (New) The processor of Claim 9, wherein the first memory controller sends a signal to the microengine when the first memory controller completes the requested access.

11. (New) The processor of Claim 9, wherein the second thread of the microengine is operable to request access to a second memory coupled to the second memory controller, and a third thread of the microengine operates while the first and second threads wait for data from the first and second memories.

12. (New) The processor of Claim 11, wherein the third thread accesses the first bus interface while the first and second threads wait for data from the first and second memories.

13. (New) The processor of Claim 11, wherein a fourth thread of the microengine processes data in a data path of the network processor.

14. (New) The processor of Claim 6, wherein the microengines access either the first or second memory controllers based on characteristics of data.

15. (New) The processor of Claim 6, wherein the bus interface is coupled to first and second network devices, the bus interface being operable to receive a plurality of packets from the first and second network devices, the microengines being operable to process the packets in parallel.

16. (New) The processor of Claim 15, being operable to process each packet independently.

17. (New) The processor of Claim 15, wherein the bus interface stores transmit, receive and interrupt flags for each network device.

18. (New) The processor of Claim 6, wherein the bus interface comprises a first-in-first-out memory operable to store packet data and packet status from the bus.

19. (New) The processor of Claim 18, wherein the bus interface is operable to issue N requests for packet data and packet status to a network device coupled to the bus, each request for M bytes; receive packet data and packet status and store received packet data and packet status in an element of the first-in-first-out memory; detect an end-of-packet indicator; determine if the last M received bytes contain data or packet status; and if the last M received bytes contain packet data, issue another request of M bytes to retrieve packet status.

20. (New) The processor of Claim 6, wherein the bus interface comprises a hash unit operable to process hash requests from the microengines.

21. (New) The processor of Claim 6, wherein the bus interface comprises a plurality of state machines operable to transfer data to and from registers in the microengines.